

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for transferring data comprising:

performing a ~~first~~ synchronization operation wherein the ~~first~~ synchronization operation uses a first processing thread to copy ~~copies~~ a first file from a first ~~memory~~ data storage to a second ~~memory~~ data storage, and a second processing thread to copy a second file from the first ~~memory~~ data storage to the second ~~memory~~ data storage, wherein the first file is copied substantially concurrently with the copying of the second file, and wherein the first operation results in a first copied file and a second copied file in the second ~~memory~~ data storage; and

performing a ~~second~~ real-time replication operation wherein the ~~second~~ real-time replication operation updates the first copied file and the second copied file in ~~a predetermined~~ an order determined at least in part by an order in which changes were made to the first file and the second file, respectively, as stored in the first data storage.
2. (Cancelled)
3. (Cancelled)
4. (Original) The method of claim 1, wherein the first file and the second file are copied regardless of order.
5. (Cancelled)

6. (Original) The method of claim 1, wherein a first command associated with the first operation can be processed by a first thread or a second thread, and a second command associated with the second operation can be processed by the second thread.

7. (Currently amended) The method of claim 1, wherein the copying of the first file is associated with a first connection between the first ~~memory~~ data storage and the second ~~memory~~ data storage, and the copying of the second file is associated with a second connection between the first ~~memory~~ data storage and the second ~~memory~~ data storage.

8. (Currently amended) A system for transferring data comprising:

a processor;

a first ~~memory~~ data storage coupled to the processor, wherein the first ~~memory~~ data storage is associated with a first file and a second file; and

wherein the processor is configured to perform a ~~first~~ synchronization operation wherein the ~~first~~ synchronization operation uses a first processing thread to copy ~~copies~~ the first file from the first ~~memory~~ data storage to a second ~~memory~~ data storage, and a second processing thread to copy a second file from the first ~~memory~~ data storage to the second ~~memory~~ data storage, wherein the first file is copied substantially concurrently with the copying of the second file, and wherein the first operation results in a first copied file in the second ~~memory~~ data storage and a second copied file in the second ~~memory~~ data storage; and also configured to perform a ~~second~~ real-time replication operation wherein the ~~second~~ real-time replication operation updates the first copied file and the second copied file in a ~~predetermined~~ an order determined at least in part by an order in which changes were made to the first file and the second file, respectively, as stored in the first data storage.

9. (Currently amended) A method for transferring data associated with a real-time data replication system comprising:

providing a ~~first~~ main thread, wherein the ~~first~~ main thread can process a ~~first~~ synchronization type of command and a ~~second~~ dynamic replication type[[s]] of command[[s]];

providing a ~~second~~ synchronization thread, wherein the ~~second~~ synchronization thread can process the ~~first~~ synchronization type of command but not the dynamic replication type of command;

wherein the synchronization thread is configured to process a command of the synchronization type substantially concurrently with the processing a first command by the [[first]] main thread of a command of the synchronization type and to not process a command of the synchronization type at a time when the main thread is processing a command of the dynamic replication type and a second command by the second thread, wherein the first and second commands are associated with the first type of command.

10. (Cancelled)

11. (Cancelled)

12. (Cancelled)

13. (Cancelled)

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Currently amended) A system for transferring data associated with a real-time data replication system comprising:

a processor configured to provide a first main thread, wherein the first main thread can process a first synchronization type of command and a ~~second~~ dynamic replication type[[s]] of command[[s]]; also configured to provide a ~~second~~ synchronization thread, wherein the ~~second~~ synchronization thread can process the first synchronization type of command but not the dynamic replication type of command; [[and]] wherein the processor also synchronization thread is configured to process a command of the synchronization type substantially concurrently with the processing by the main thread of a command of the synchronization type and to not process a command of the synchronization type at a time when the main thread is processing a command of the dynamic replication type ~~transfer a first command by the first thread and a second command by the second thread, wherein the first and second commands are associated with the first type of command~~; and

a memory coupled to the processor for providing the processor with instructions.

19. (Currently amended) A computer program product for transferring data, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

performing a first synchronization operation wherein the first synchronization operation uses a first processing thread to copy ~~copies~~ a first file from a first memory data storage to a second ~~memory~~ data storage, and a second processing thread to copy a second file from the first

~~memory data storage~~ to the second ~~memory data storage~~, wherein the first file is copied substantially concurrently with the copying of the second file, and wherein the first operation results in a first copied file and a second copied file in the second ~~memory data storage~~; and performing a ~~second~~ real-time replication operation wherein the ~~second~~ real-time replication operation updates the first copied file and the second copied file in a ~~predetermined~~ an order determined at least in part by an order in which changes were made to the first file and the second file, respectively, as stored in the first data storage.

20. (Currently amended) A computer program product for transferring data associated with a real-time data replication system, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

providing a ~~first~~ main thread, wherein the ~~first~~ main thread can process a ~~first~~ synchronization type of command and a ~~second~~ dynamic replication type[[s]] of command[[s]];

providing a ~~second~~ synchronization thread, wherein the ~~second~~ synchronization thread can process the ~~first~~ synchronization type of command but not the dynamic replication type of command;

wherein the synchronization thread is configured to process a command of the synchronization type substantially concurrently with the processing a ~~first command~~ by the [[first]] main thread of a command of the synchronization type and to not process a command of the synchronization type at a time when the main thread is processing a command of the dynamic replication typeand a second command by the second thread, wherein the first and second commands are associated with the first type of command.

21. (New) The method of claim 9 wherein the main thread, the synchronization thread, or both process commands from a kernel cache.

22. (New) The method of claim 21 wherein the synchronization thread skips commands in the kernel cache that have been or are being processed by other threads until it finds a synchronization command that has not yet been and is not currently being processed by another thread.
23. (New) The method of claim 21 wherein when the synchronization thread does not move ahead of the main thread unless the main thread is performing a synchronization command.
24. (New) The method of claim 21 wherein if a synchronization thread does not encounter synchronization commands in the kernel cache, the synchronization thread closes after a time interval.
25. (New) The method of claim 9 wherein the main thread does not process dynamic replication types of commands unless all synchronization threads that are executing synchronization commands are completed.